Anroid Logging System

William.L

wiliwe@gmail.com

Date: 2011-07-21

Outline

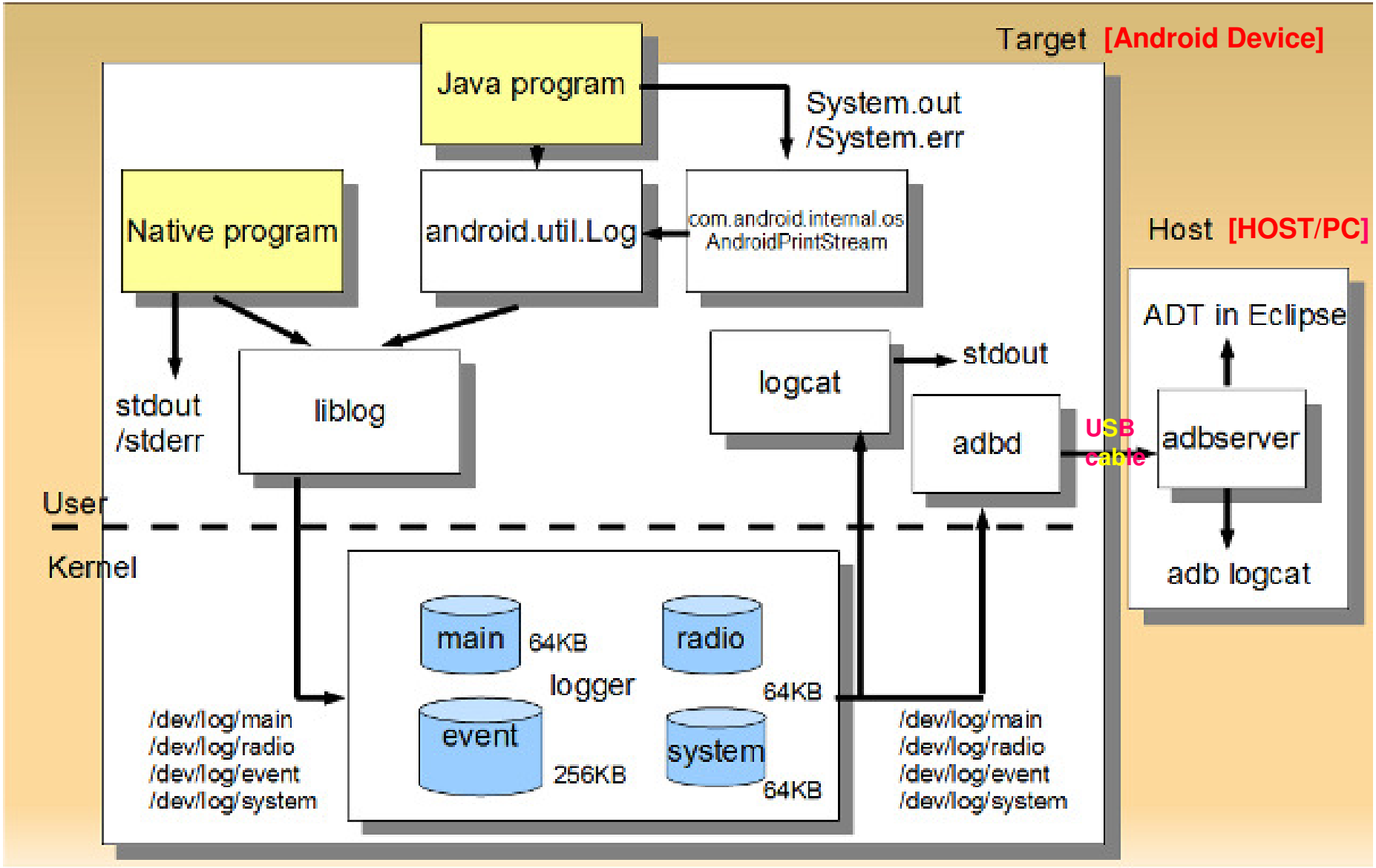
- Overview of Android Loggin System
- Log from Java program
- Log from Native C/C++ program
- How to read log?
- Tips

Overview of Android Logging System

What is Android Logging System?

- Provide a mechanism for collecting and viewing system debug output
- Logs from various applications and portions of the system are collected in a series of circular buffers, which then can be viewed and filtered by the **logcat** command

Overview of Logging System



Introduction

- The logging system consists of
  - A **kernel driver** and **kernel buffers** for storing log messages
    - HoneycombMR2Src/kernel/drivers/staging/android/**logger.c**
    - Create "/**dev/log**" folder in **handle_device_event()** of AndroidSrc/system/core/init/**devices.c**
  - C/C++/Java APIs and classes
    - For making log messages
    - For accessing the log messages
  - **logcat**, the command for viewing log messages
    - AndroidSrc/system/core/logcat/
  - Ability to view and filter the log messages from the host machine (via **Eclipse-ADT** or **DDMS**)

Log device files

- 4 channels, each have a Ring/Circular Buffer
  - /dev/log/**radio** – radio&phone-related messages (64KB)
  - /dev/log/**events** – system/hardware events (256KB)
  - /dev/log/**system** –framwork or low-level system messages (64KB)
  - /dev/log/**main** – everything else (64KB)
  - The maximum log message size of each channel is specified in kernel driver(logger.c)
- File permission of each(radio/events/system/main) is **0662 (rw-rw-w)**
  - owner/group RW, other Write only
  - owner=root, group=log

```
william@will-i-am:~$ adb shell
# ls -ls /dev/log
total 0
crw-rw--w- root     log       10,  36 2009-01-29 18:50 events
crw-rw--w- root     log       10,  37 2009-01-29 18:50 main
crw-rw--w- root     log       10,  35 2009-01-29 18:50 radio
crw-rw--w- root     log       10,  34 2009-01-29 18:50 system
```

  - Anyone can Write logs, root or log group can Read them

Android Debug Bridge

- ADB client
  - Runs on your development machine(Host).
  - You can invoke a client from a shell by issuing an “**adb**” command.
  - Other Android tools such as the **ADT** plugin and **DDMS** also create adb clients.
- ADB server (**adbserver**)
  - Runs on your development machine(Host).
  - The server manages communication between the client and the adb daemon running on an emulator or device.
- ADB daemon (**adbd**)
  - Runs on each Android emulator or Android device instance.

Log from Java program

Classes used for Logging

- android.util.Log class
- System.out / System.err

android.util.Log

- Static methods
  - AndroidSrc/frameworks/base/core/java/android/util/Log.java

Log.v (String tag, String msg)	Verbose message
Log.d (String tag, String msg)	Debugging message
Log.i (String tag, String msg)	Info message
Log.w (String tag, String msg)	Warrning message
Log.e (String tag, String msg)	Error message

Example :

```
/* In Java code, add the following codes */

import android.util.Log;

class CCLLAASS {
   static String TAG="tagName";

   public MethodXX() {
      Log.v(TAG, "Debugging messages you want");
   }
}
```

System.out / System.err (1/2)

- System.out/System.err output to Android log
  - zygoteInit() {
    System.setOut(AndroidPrintStream);
    System.setErr(AndroidPrintStream); }
    - AndroidSrc/frameworks/base/core/java/com/android/internal/os/RuntimeInit.java
  - com.android.internal.os.AndroidPrintStream (which derives from **LoggingPrintStream** which derives from **PrintStream**)
    - AndroidSrc/frameworks/base/core/java/com/android/internal/os/AndroidPrintStream.java

System.out / System.err (2/2)

- How to identify instance of System.out/System.err?
  - System.out.println("System.out="+System.out.toString())
  - System.err.println("System.err="+System.err.toString())

Example :

```
/* Add the System.out and System.err statements in the constructor of MountService.java */
class MountService {
   MountService() {
      ....
      System.out.println("System.out's instance is "+System.out.toString());
       System.err.println("System.err's instance is "+System.err.toString());
      ....
   }
}
```

```
01-29 21:54:41.270: DEBUG/NetworkManagmentService(123): Registering observer
01-29 21:54:41.280: INFO/MountService(123): Enter MountService()
01-29 21:54:41.280: INFO/MountService(123): java.lang.Throwable
01-29 21:54:41.280: INFO/MountService(123):     at com.android.server.MountService.<init>(MountService.java:1175)
01-29 21:54:41.280: INFO/MountService(123):     at com.android.server.ServerThread.run(SystemServer.java:322)
01-29 21:54:41.280: INFO/System.out(123): System.out's instance is com.android.internal.os.AndroidPrintStream@40713610
01-29 21:54:41.280: WARN/System.err(123): System.err's instance is com.android.internal.os.AndroidPrintStream@407137d8
01-29 21:54:41.280: INFO/SystemServer(123): Throttle Service
```

Log from Native C/C++ program

Library for Logging

- Use **liblog** library
- Include **<android/log.h>** header
- **<cutils/log.h>**(libcutils) header could be used
  - This header includes <android/log.h> eventually
- **__android_log_print** macro(defined in **liblog**) is the actual worker behind **LOGI**[**V**/**D**/**W**/**E**] functions

Example :

```
#define LOGI(...) \
__android_log_print (ANDROID_LOG_INFO,LOG_TAG,__VA_ARGS__)
```

Usage :

```
LOGI("i=%d, name=%s\n", i, name);
```

Log From Native Program (1/2)

- Log functions

LOGV (String msg)	Verbose message.
LOGD (String msg)	Debugging message
LOGI (String msg)	Info message
LOGW (String msg)	Warrning message
LOGE (String msg)	Error message

Example :

```
/* In C/C++ code, add the following codes*/
#define LOG_TAG “tagName”
#include <cutils/log.h>

LOGV(“Debugging messages you want”);
```

Log From Native Program (2/2)

- It must add following definition BEFORE the header "#include <cutils/log.h>"
  - #undef NDEBUG : Enable **LOGV/LOGI/LOGD**
  - #define LOG_NDEBUG 0 : Enable **LOGV**
  - #define LOG_NIDEBUG 0 : Enable **LOGI**
  - #define LOG_NDDEBUG 0 : Enable **LOGD**
  - #define LOG_TAG “String-You-Want"
- Because all the above are defined in **<cutils/log.h>**, if the **define** is put after the header including statment, it will show “**redefined**” compile warning and **define** will not take effect

How to read log?

Logcat Command

- "**logcat**" command runs on Android device
- Use the command to run 'logcat' command on the **remote** Android device : "**adb shell logcat**"

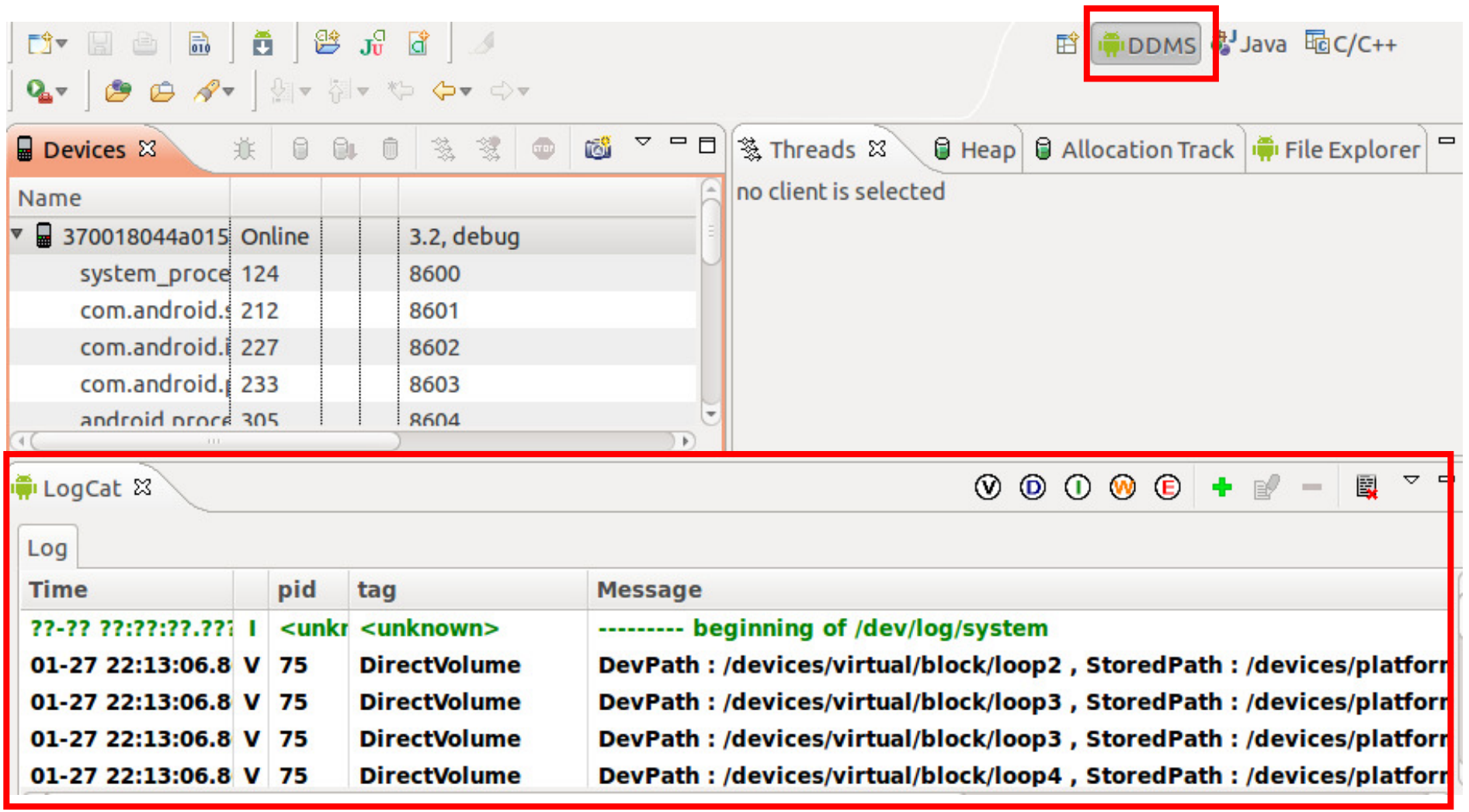
```
william@will-i-am:~/pbj30.mr2$ adb shell logcat
--------- beginning of /dev/log/system
I/Vold    (   75): Vold 2.1 (the revenge) firing up
D/Vold    (   75): USB mass storage support is not enabled in the kernel
D/Volume  (   75): Volume usbdrive state changing -1 (Initializing) -> 0 (No-Media)
D/Volume  (   75): Volume sdcard state changing -1 (Initializing) -> 0 (No-Media)
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop0 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop0 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop1 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop1 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop2 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop2 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop3 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop3 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop4 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop4 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop5 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop5 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop6 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop6 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop7 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop7 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
```

ADB Logcat

- Command : **adb logcat**

```
william@will-i-am:~/pbj30.mr2$ adb logcat
--------- beginning of /dev/log/system
I/Vold    (   75): Vold 2.1 (the revenge) firing up
D/Vold    (   75): USB mass storage support is not enabled in the kernel
D/Volume  (   75): Volume usbdrive state changing -1 (Initializing) -> 0 (No-Media)
D/Volume  (   75): Volume sdcard state changing -1 (Initializing) -> 0 (No-Media)
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop0 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop0 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop1 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop1 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop2 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop2 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop3 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop3 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop4 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop4 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop5 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop5 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop6 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop6 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop7 , StoredPath : /devices/platform/tegra-ehci.2/usb2
V/DirectVolume(   75): DevPath : /devices/virtual/block/loop7 , StoredPath : /devices/platform/sdhci-tegra.2/mmc_host/mmc1
```

Logcat pane in ADT, Eclipse



Tips

- Dumping stack trace
- logwrapper
- Log at ‘init’ process
- Support Non built-in ADB USB VID

Dumping stack trace (1/2)

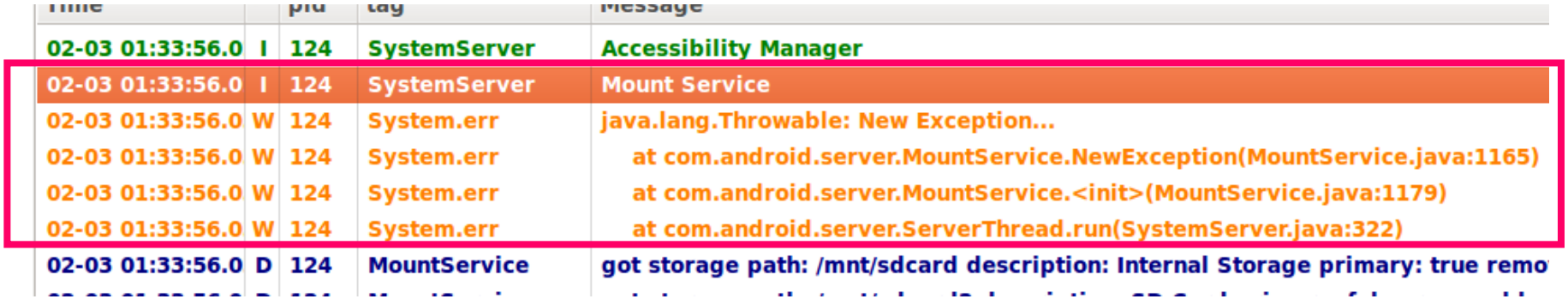
- 3 arguments methods in **android.util.Log** class
  - Ex: Log.e(String tag, String msg, **new Throwable()**)
- **Throwable.printStacktrace**() also works
  - Dump to System.err

Dumping stack trace (2/2)

Example for **Throwable.printStackTrace** (**MountService.java**) :

```
class MountService extends IMountService.Stub implements INativeDaemonConnectorCallbacks
{
    public static void NewException() throws Throwable
    {
        throw new Throwable("New Exception...");
    }

    public MountSerivce(Context context) {
      …
      try {
          NewException();
      } catch (Throwable e) {
          // Prints this throwable and its backtrace to the
          // standard error stream.
          e.printStackTrace();
      }
      ...
  }
  …
}
```



Time		pid	tag	Message
02-03 01:33:56.0	I	124	SystemServer	Accessibility Manager
02-03 01:33:56.0	I	124	SystemServer	Mount Service
02-03 01:33:56.0	W	124	System.err	java.lang.Throwable: New Exception...
02-03 01:33:56.0	W	124	System.err	at com.android.server.MountService.NewException(MountService.java:1165)
02-03 01:33:56.0	W	124	System.err	at com.android.server.MountService.<init>(MountService.java:1179)
02-03 01:33:56.0	W	124	System.err	at com.android.server.ServerThread.run(SystemServer.java:322)
02-03 01:33:56.0	D	124	MountService	got storage path: /mnt/sdcard description: Internal Storage primary: true remo

logwrapper

- Redirects stdout(like **printf**)/stderr to Android Logging system
- Usage
  - "logwrapper Executable", and use "logcat" to watch logs as usual
  - Ex : "logwrapper ObbFile_test"

```
# ObbFile_test
Running main() from gtest_main.cc
[==========] Running 2 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 2 tests from ObbFileTest
[ RUN      ] ObbFileTest.ReadFailure
[       OK ] ObbFileTest.ReadFailure
[ RUN      ] ObbFileTest.WriteThenRead
[       OK ] ObbFileTest.WriteThenRead
[----------] Global test environment tear-down
[==========] 2 tests from 1 test case ran.
[  PASSED  ] 2 tests.
```

Executing **without** 'logwrapper'

```
I/ObbFile_test( 2633): Running main() from gtest_main.cc
I/ObbFile_test( 2633): [==========] Running 2 tests from 1 test case.
I/ObbFile_test( 2633): [----------] Global test environment set-up.
I/ObbFile_test( 2633): [----------] 2 tests from ObbFileTest
I/ObbFile_test( 2633): [ RUN      ] ObbFileTest.ReadFailure
W/ObbFile ( 2634): attempt to read from invalid fd
I/ObbFile_test( 2633): [       OK ] ObbFileTest.ReadFailure
I/ObbFile_test( 2633): [ RUN      ] ObbFileTest.WriteThenRead
I/ObbFile_test( 2633): [       OK ] ObbFileTest.WriteThenRead
I/ObbFile_test( 2633): [----------] Global test environment tear-down
I/ObbFile_test( 2633): [==========] 2 tests from 1 test case ran.
I/ObbFile_test( 2633): [  PASSED  ] 2 tests.
I/logwrapper( 2633): ObbFile test terminated by exit(0)
```

Executing **with** 'logwrapper'

Log at init process

- The first process, '**init**', does not use Android Logging System.
- '**init**' writes log to (the same node as) '/dev/kmsg'
  - The same way as '**printk**()'
- Add a command in **init.rc** to write Android logs to kernel logging file, **/dev/kmsg**
  - Command : service logcat /system/bin/logcat -f /dev/kmsg oneshot
  - Watch logs : run "**adb shell dmesg**" on the host
  - Shortpoint : duplicated store of Android log
- To save output messages of logcat
  - **logcat -f fileName**

Support Non built-in ADB USB VID (1/2)

- ADB built-in USB VID
  - http://developer.android.com/guide/developing/device.html#VendorIds
- Solution-1 : Append the new USB VID into the **adb_usb.ini** file
  - Commands (executing on the host, e.g.PC/NB) :
    - Create the folder/file '~/**.android/adb_usb.ini**' if it does not exist
      - '**adb**' command reads and checks content of this file **each time** it is executed
    - echo "New-Vendor-ID" >> ~/.android/adb_usb.ini
    - sudo -s "adb kill-server;adb start-server"

Example (**Lenovo** VID : **0x17EF**) :

```
/* It can watch the VID of an Android device
   using 'lsusb' command under the host */
#> echo "0x17EF" >> ~/.android/adb_usb.ini
#> sudo -s "adb kill-server;adb start-server"
```

Support Non built-in ADB USB VID (2/2)

- Solution-2 : Build a new '**adb**' tool supporting new VID
  - In AndroidSrc/system/core/adb/usb_vendors.c
  - #define **VENDOR-NAME** Vendor-ID
  - Add an entry with new **VENDOR-NAME** in variable builtInVendorIds[] and then compile '**adb**' sources
  - Built new '**adb**' exectuable is under the folder : **out/host/linux-x86/bin/**

Ex - In **AndroidSrc/system/core/adb/usb_vendors.c** :

```
// Lenovo's USB Vendor ID
#define VENDOR_ID_LENOVO    0x17EF

/** built-in vendor list */
int builtInVendorIds[] = {
                        ..... ,
            VENDOR_ID_LENOVO
};
```

Reference

- http://elinux.org/Android_Logging_System
- Android Logging system slide - http://blog.kmckk.com/archives/2936958.html
- logwrapper - http://blog.kmckk.com/archives/2918551.html
- Print Call Stack - http://blog.kmckk.com/archives/2902690.html